

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Maja Zafran

Prototip mobilne aplikacije mStudis

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2018

Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Na FRI za študijsko področje uporabljamo informacijski sistem Studis. Študentom bi bila zelo dobrodošla mobilna aplikacija, ki bi študentom omogočila prijavo in odjavo na izpit oz. iz izpita. Opredelite funkcionalne zahteve za mobilno aplikacijo mStudis, ki naj bo namenjena tako študentom kot tudi pedagogom. Na podlagi funkcionalnosti nato razvijte prototip mobilne aplikacije mStudis za operacijski sistem iOS.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Predstavitev problema	2
1.2	Predstavitev področja uporabe mobilne aplikacije mStudis . . .	3
1.3	Cilji diplomske naloge	4
2	Uporabljene tehnologije	5
2.1	Razvojno okolje Xcode	5
2.2	Programski jezik Swift	6
2.3	App Store in TestFlight	6
2.4	Uporabljene knjižnice	7
2.5	Shranjevanje podatkov	8
3	Razvoj mobilne aplikacije	11
3.1	Prototip mobilne aplikacije mStudis	11
3.2	Arhitektura prototipa mobilne aplikacije mStudis	16
3.3	Implementacija prototipa mobilne aplikacije mStudis	17
3.4	Potisna obvestila	26
4	Zaključek	29
4.1	Možnosti za nadaljevanje razvoja	30

Povzetek

Naslov: Prototip mobilne aplikacije mStudis

Avtor: Maja Zafran

Glavni cilj diplomske naloge je razvoj prototipa mobilne aplikacije mStudis, s katero bodo študentje ter profesorji in asistenti Fakultete za računalništvo in informatiko lažje spremljali izpitne roke in dosežene rezultate. Študentje se bodo lahko preko mobilne aplikacije prijavili ali odjavili od izpitov. Prav tako bodo lahko preko aplikacije videli dosežene rezultate (ocene), o katerih bodo obveščeni preko potisnih obvestil. Profesorji in asistenti bodo lahko preko nje videli seznam prijavljenih študentov za določen izpit, posameznega študenta odjavili od izpita (mu vrnili prijavnico) ter vpisovali dosežene točke in oceno izpita.

V diplomski nalogi sem predstavila tehnologije, ki sem jih uporabila pri razvoju in distribuiranju mobilne aplikacije mStudis, aplikacijo samo ter njeno arhitekturo in primere uporabe. Za namen testiranja mobilne aplikacije sem postavila tudi testno bazo in jo napolnila s testnimi podatki. Predstavila sem tudi implementacijo potisnih obvestil, ki sem jo dosegla s pomočjo zunanje knjižnice in Applove storitve za potisna obvestila. Na koncu sem predstavila tudi možnosti za nadaljnji razvoj aplikacije.

Ključne besede: mobilna aplikacija, Swift, operacijski sistem iOS.

Abstract

Title: Prototype of mStudis mobile application

Author: Maja Zafran

The main goal of this thesis is the development of the mStudis mobile application prototype, which will help students, professors, and assistants of the Faculty of Computer and Information Science keep track of their upcoming exams and their results. Through the mobile application, students will be able to apply to upcoming exams or cancel their applications. They will be able to see their results and grades and be informed about them through push notifications. Professors and assistants will be able to get a list of students who applied to each exam, cancel a specific student's application, and for each student, insert points and grades.

In my thesis, I present technologies, which I used while developing and distributing the mStudis mobile application prototype, the application itself, its architecture, and use cases. For testing purposes I made a test database and filled it with test data. The application includes an implementation of push notifications, which I achieved with the help of an outside framework and Apple push notification services. At the end, I present possibilities for further improvements.

Keywords: mobile application, Swift, iOS operating sistem.

Poglavje 1

Uvod

Telefonov dandanes ne uporabljamo več samo za klicanje in pošiljanje sporočil. Postajajo čedalje pametnejši, sposobnejši, povečuje se njihova uporabnost in uporaba. Število mobilnih naprav je že preseglo število ljudi na svetu[24].

Razvoj pametnih telefonov je pripeljal do tega, da ljudje želimo do vseh informacij priti v čim krajšem času ter biti čim hitreje nagrajeni za svoje delo.

Uporabniki pametnih telefonov, med njimi predvsem mladi, so navajeni, da imajo preko njih dostop do celega svojega življenja. Na telefonih lahko brskajo po internetu, spremljajo socialna omrežja, se pogovarjajo s prijatelji, spoznavajo nove ljudi, igrajo igrice, vodijo posel itd. Pametni telefoni so praktični, saj jih imamo konstantno na dosegu roke, kjerkoli in kadarkoli.

Zato je tudi normalno, da se čedalje več podjetij in ustanov odloča za mobilne aplikacije. Preko njih ostanejo v kontaktu s svojimi strankami oziroma člani, jih obveščajo o spremembah in novostih ter jim omogočajo lažjo komunikacijo z zaposlenimi. Prav tako postajajo mobilne aplikacije, ki so namenjene interni komunikaciji v podjetju ali ustanovi, čedalje bolj popularne. Omogočajo lažjo komunikacijo med zaposlenimi ter pripomorejo k dvigu hitrosti in kvalitete dela.

1.1 Predstavitev problema

Izpitno obdobje je stresen čas za vse na fakulteti, zato je dobro, da se jim ga čim bolj olajša. Študentje imajo polne glave priprav na izpite ter čakanja na ocene, profesorjem in asistentom pa se mudi sestaviti in popraviti izpite.

V pripravi na izpitno obdobje se za vsak predmet razpišejo izpitni roki. Ponavadi sta v zimskem in poletnem izpitnem obdobju razpisana dva izpitna roka, v jesenskem pa eden za vsak predmet. Študentje se na te roke lahko prijavijo preko spletne strani Studis[16]. Trenutno je za to potrebno imeti dostop do računalnika ali pa se prijaviti preko spletnega brskalnika na telefonu. Študentu vsakokratno vpisovanje spletnega naslova in podatkov za prijavo vzame veliko časa, hkrati pa stran ni najboljše optimizirana za prikaz na telefonih.

Eno izmed bolj stresnih obdobij za vsakega študenta je zadnjih nekaj ur pred izpitom. Živci so takrat na vrhuncu in celo telo je v pričakovanju izpitnih vprašanj. Takrat se tudi veliko študentov odloči za odjavo od izpita ali pa samo želijo preveriti, kdaj in kje bo potekal. Za to se morajo ponovno prijaviti na spletni strani, kar zopet vzame dragoceni čas in preusmerja koncentracijo v stran od učenja za izpit.

Najtežje pa je na koncu čakati na rezultate. Veliko študentov po koncu izpita sprašuje profesorja, kdaj bodo izpiti popravljene oziroma kdaj bodo vpisane ocene. Seveda se redkokdaj zgodi, da so popravljene točno na predviden dan. Popravljanje je namreč odvisno od profesorja ter drugih obveznosti, ki jih ima v tistem obdobju. Tako študentom ne preostane drugega, kot da vsakih nekaj minut osvežujejo spletno stran z ocenami ali pa se zanesejo na sošolce, da jim bodo povedali, ko bodo ocene vpisane.

Izpitno obdobje ni naporno le za študente, temveč tudi za profesorje in assistente. Sestavljanje izpitov, njihovo izvajanje in popravljanje lahko vzamejo zelo veliko časa in energije. Izpiti si sledijo drug za drugim, pedagogi pa morajo za vsak izpit priti do računalnika, da pogledajo, kje imajo naslednjega in koliko študentov je nanj prijavljenih. Na izpite po navadi poleg izpitnih pol prinesejo še seznam prijavljenih študentov, kjer si zapišejo, kdo manjka in

ali se želi kdo odjaviti od izpita. Po izpitu je seveda vse to potrebno prepisati še v elektronski indeks.

Profesorji in asistenti so tudi vedno pod pritiskom, da čim hitreje popravijo izpite in študente seznanijo z njihovimi ocenami. Že ob koncu izpita študentje želijo vedeti, kdaj bodo objavljene ocene. Študentje tudi po e-pošti sprašujejo o doseženih točkah in oceni. Profesorju oziroma asistentu že popravljanje izpitov vzame veliko časa, potem pa je treba še vse dosežene točke in ocene vpisati v elektronski indeks. Ob popravljanju izpitov pedagogi niso vedno v bližini računalnika, da bi lahko ocene sproti vpisovali, kar še dodatno podaljša čas obveščanja študentov.

Prav zaradi vsega zgoraj opisanega bi bilo dobro, da bi imeli študentje in profesorji možnost, da vsaj nekaj – če ne kar vseh – teh dejanj opravijo preko telefona, ki ga imajo zmeraj pri sebi. In to je tudi tema pričujoče diplomske naloge.

1.2 Predstavitev področja uporabe mobilne aplikacije mStudis

V diplomski nalogi bom opisala prototip mobilne aplikacije mStudis, ki sem jo razvila kot pomoč študentom in profesorjem pred in po izpitnih rokih. Mobilna aplikacija mStudis tako vsem uporabnikom omogoča dostop do elektronskega indeksa in tistih podatkov o aktualnih izpitnih rokih, ki so zanje pomembni.

Mobilna aplikacija je primarno namenjena študentom in pedagogom Fakultete za računalništvo in informatiko. Narejena je za lažje spremljanje izpitnih rokov in ocen, zato vsebuje temu primerne podatke. Za namen testiranja mobilne aplikacije sem postavila tudi spletni strežnik in testno bazo. Uporabljene podatke in bazo ter razloge za posamezne odločitve sem predstavila v tretjem poglavju diplomske naloge. Predstavila sem tudi primere uporabe in arhitekturo mobilne aplikacije.

Aplikacija je trenutno v prototipni različici in jo je mogoče dobiti le preko

povabila za testiranje. Za vsak pogled v mobilni aplikaciji sem predstavila, kakšne možnosti in izbiro ima na voljo posamezni uporabnik.

Zaradi obširnosti tematike sem se osredotočila le na izdelavo mobilne aplikacije za iOS operacijski sistem. V zaključku diplomske naloge sem predstavila še svoje predloge za izboljšavo mobilne aplikacije.

1.3 Cilji diplomske naloge

Cilj diplomske naloge je izdelati prototip mobilne aplikacije mStudis, ki bo v veliki meri podobna spletnemu vmesniku Studis[16]. Mobilna aplikacija je namenjena študentom ter profesorjem in asistentom na fakulteti za računalništvo in informatiko.

Njena glavna funkcionalnost je spremljanje aktualnih izpitov.

Študentje se bodo lahko preko mobilne aplikacije prijavljali k posameznemu izpitu, se po potrebi od njega tudi odjavili, po opravljenem izpitu pa bodo lahko na enostaven način obveščeni o doseženem rezultatu. Ves čas jim je na vpogled tudi elektronski indeks z vsemi vpisanimi ocenami.

Profesorji in asistenti pa se bodo preko mobilne aplikacije lahko seznanili, koliko in kateri študentje so se na posamezni izpit prijavili ali pa študenta odjavili od izpita. Po izpitu bodo lahko preko nje vpisali še dosežene točke in oceno izpita.

Poglavje 2

Uporabljene tehnologije

V tem poglavju bom predstavila tehnologije, ki sem jih uporabila pri razvoju prototipa mobilne aplikacije mStudis. Mobilno aplikacijo sem razvijala v razvojnem okolju Xcode in s programskim jezikom Swift, ki je trenutni standard za razvoj aplikacij za operacijski sistem iOS. Predstavila bom tudi Applovo platformo za testiranje mobilnih aplikacij TestFlight. Na koncu bom predstavila še knjižnico OneSignal, ki sem jo uporabila pri razvoju potisnih obvestil ter uporabljene načine shranjevanja podatkov.

2.1 Razvojno okolje Xcode

Xcode je integrirano razvojno okolje, namenjeno razvijanju aplikacij za Apple naprave kot so iPhone, iPad, Apple TV, Apple Watch in Mac računalnik. Izšel je leta 2003 in vsebuje zbirko orodij za razvoj aplikacij. Poleg urejevalnika in prevajalnika kode vsebuje tudi simulator za različne Apple naprave. Trenutno (september 2018) je v App Store na voljo različica 9.4.1. Apple je letos na svoji konferenci naznanil izid razvojnega okolja Xcode 10, ki je trenutno razvijalcem na voljo samo v beta različici[8]. Uporabnikom Mac računalnikov je v App Store najnovejša produkcijska različica razvojnega okolja Xcode na voljo brezplačno . Za pridobitev beta različice pa mora imeti razvijalec pri Applu ustvarjen razvijalski račun[19].

Razvojno okolje Xcode je prilagojeno za več programskih jezikov, kot so na primer Swift, Objective-C, C, Java. Do leta 2014 je bil glavni programski jezik za razvoj iOS aplikacij Objective-C. Tega leta pa je Apple izdal nov programski jezik Swift, ki je postal neformalni standard razvojnega okolja Xcode [20, 22, 21].

2.2 Programski jezik Swift

Programski jezik Swift je objektni programski jezik, ki se uporablja za razvijanje aplikacij za Apple naprave. Prva različica je bila objavljena leta 2014, leto kasneje pa je z različico 2.2 postal odprtokoden programski jezik, ki podpira iOS, OS-X in Linux. Najnovejša produkcijska različica je 4, v kateri je bil narejen tudi prototip mobilne aplikacije mStudis.

Programski jezik Swift je naslednik jezika Objective-C. Zasnovan je bil na treh načelih: da je varen, moderen in zmogljiv. Zaradi svojih prednosti je postal de facto programski jezik za ustvarjanje privlačnih in odzivnih aplikacij.

Programski jezik Swift ima v primerjavi z jezikom Objective-C več prednosti. Vgrajenih ima, na primer, kar nekaj varoval za preprečevanje napak in lažjo detekcijo le-teh, zaradi česar je odpravljanje napak enostavnejše.

Modernost programskega jezika Swift izhaja iz intuitivne sintakse, ki se bere podobno kot angleščina. V primerjavi z Objective-C potrebuje za isto funkcionalnost tudi manj kode, kar olajša vzdrževanje programske kode[23, 15, 26, 14].

2.3 App Store in TestFlight

App Store je digitalna platforma podjetja Apple za distribucijo mobilnih aplikacij za iOS operacijske sisteme. Nastala je leta 2008, danes pa vsebuje že preko 2 milijona aplikacij[25].

App Store razvijalcem omogoča, da svojo mobilno aplikacijo objavijo na

platformi, s čimer postane dostopna oziroma jo lahko uporabljajo vsi uporabniki iOS naprav. Vsaka naložena aplikacija mora naprej skozi Applov odobritveni proces. Med tem procesom Apple preveri, ali aplikacija deluje po pričakovanjih, je zanesljiva in ne vsebuje žaljive vsebine. Šele po uspešno prestatem testu je aplikacija objavljena na App Store. V nasprotnem primeru dobi razvijalec sporočilo z vrsto napake in razlogom za zavrnitev. Po odpravi napak se lahko aplikacija ponovno namesti na App Store oziroma pošlje skozi odobritveni proces[2, 3].

Za razvijalce aplikacij je velikokrat pomembno, da dobijo konstruktivno povratno informacijo o svoji aplikaciji še preden jo javno objavijo v App Store. Tega se je zavedal tudi Apple, zato je v ta namen vzpostavil platformo TestFlight. Razvijalec na platformi objavi svojo beta aplikacijo in navede osebe (t. i. testerje), za katere želi, da njegovo aplikacijo prvi preizkusijo. Testerji prejmejo nato preko e-pošte povabilo, da si novo beta aplikacijo naložijo in jo preizkusijo. Ob vsaki novi posodobitvi aplikacije dobijo testerji potisno sporočilo, da je izšla nova različica, in možnost, da si jo naložijo.

TestFlight omogoča interno in eksterno testiranje aplikacije. Razvijalci in člani razvijalske ekipe imajo dostop do vseh beta različic, ki so na voljo za testiranje. Skupaj imajo lahko do 25 računov, pri čemer lahko vsak račun aplikacijo testira na 30 različnih napravah. Naslednja faza testiranja je pridobitev zunanjih testerjev. Pred zunanjim testiranjem mora aplikacija prestat Applov beta odobritveni proces, kjer se preveri, ali je aplikacija v skladu z Applovimi navodili za objavo aplikacij. Nato lahko razvijalci povabijo do 10.000 ljudi za zunanje testiranje aplikacije.

Na ta način lahko testerji razvijalcu podajo koristne informacije o delovanju testne aplikacije ter predloge za njen nadaljnji razvoj[4].

2.4 Uporabljene knjižnice

Knjižnica v računalništvu opisuje zbirko podprogramov oziroma funkcij, ki so v pomoč pri razvoju programske opreme. Poznamo veliko različnih knjižnic,

ki so lahko dinamične ali statične. Večinoma se uporabljajo dinamične knjižnice, kar pomeni, da se koda knjižnice nahaja izven programa, ki knjižnico uporablja.

Knjižnice močno pomagajo programerjem pri razvoju in doseganju ciljev aplikacij, saj jim z uporabo knjižnic ni potrebno še enkrat programirati funkcij, ki so v knjižnici že implementirane. Knjižnice so lahko odprtokodne ali komercialne. V diplomski nalogi sem uporabila odprtokodno knjižnico OneSignal.

2.4.1 OneSignal

OneSignal je brezplačna knjižnica za lažjo implementacijo potisnih sporočil. Ponuja lahko in hitro implementacijo ter preprosto integracijo z različnimi platformami, kot so na primer Android, iOS, spletne strani, Unity, React Native in Xamarin. Razvijalcem nudi orodja za segmentacijo uporabnikov, avtomatično pošiljanje potisnih sporočil, orodja za poročanje itd.. Ponuja tudi veliko izbiro trženjskih orodij kot so, na primer, A/B testiranje, ciljna segmentacija trga, lokalizacija.

Knjižnica OneSignal podpira operacijske sisteme od iOS 7 naprej. Za integracijo v Xcode je napisana v programskem jeziku Swift in Objective-C[11, 12, 10].

2.5 Shranjevanje podatkov

Shranjevanje podatkov je ena izmed ključnih sestavin vsake aplikacije. Shranjevanje je lahko začasno (podatki obstajajo le dokler deluje aplikacija oziroma je naprava, v kateri teče aplikacija, aktivna) ali trajno (podatki obstajajo tudi po tem, ko aplikacija ne deluje več ali naprav ni aktivna). Obstaja veliko načinov trajnega shranjevanja podatkov, izbira pa je odvisna od količine in občutljivosti podatkov. Na odjemalcih iOS je najbolj preprost in zelo omejen način shranjevanja podatkov UserDefaults. Tega sem uporabila tudi sama za shranjevanje nastavitev in uporabniškega imena.

Večje količine podatkov se shranjujejo v za to namenjenih datotekah. Vse datoteke skupaj predstavljajo podatkovno zbirko aplikacije. Iz varnostnih razlogov lahko vsaka aplikacija dostopa le do svoje podatkovne zbirke. V podatkovnih zbirkah so podatki lahko shranjeni v različnih oblikah (formatih), kot so na primer tekstovne datoteke (npr. TXT ali CSV), relacijski podatkovni modeli (po katerih poizvedujemo z SQL). Pri aplikacijah, ki imajo opravka z večjo količino podatkov je priporočljivo, da uporabljajo tudi primeren sistem za upravljanje podatkovne zbirke (angl. Database Management System ali DBMS), ki omogoča upravljanje podatkov (definiranje, kreiranje, poizvedovanje, posodabljanje in brisanje). Sistem za upravljanje podatkovne zbirke lahko razvijalec aplikacije razvije sam ali pa uporabi eno izmed obstoječih orodij.

Med najbolj priljubljenimi Applovimi sistemi za upravljanje podatkovnih zbirk je Core Data. Njegova prednost je, da lahko shranjuje poljubne podatke v katerikoli obliki. Za shranjevanje podatkov uporablja bazo SQLite. Poleg shranjevanja, sledenja, spreminjanja in filtriranja podatkov za iOS aplikacije ponuja tudi upravljanje objektnih grafov, sledenje spremembam in še marsikaj.

Eden pomembnejših načinov shranjevanja podatkov, ki jih omogoča Apple, je sistem Keychain. Uporablja se za shranjevanje in upravljanje z občutljivimi podatki, kot so gesla, zasebni ključi, certifikati, skrite kode in podobno. Keychain sem v mobilni aplikaciji mStudis uporabila za shranjevanje gesla.

Poleg zgoraj opisanih načinov shranjevanja obstaja tudi veliko zunanjih knjižnic za shranjevanje in upravljanje podatkov, kot so RealmDB, itd[17, 18, 7, 13].

2.5.1 UserDefaults

Kot sem že omenila, je način UserDefaults namenjen shranjevanju zelo majhne količine podatkov. Z njim je mogoče shranjevati le točno določene tipe podatkov, kot so Data, String, Number, Array in Dictionary. iOS operacijski sistem za vsako aplikacijo ustvari datoteko z imenom plist, kamor potem shra-

njuje omenjene podatke. Dovoljena količina shranjenih podatkov je omejena na velikost ene datoteke, kar pomeni približno 4 GB. Kljub temu, da je 4 GB relativno velika količina podatkov, se priporoča, da se UserDefaults uporablja za shranjevanje majhnih količin podatkov. Datoteka se namreč vedno bere in zapisuje kot celota, zato se čas branja in zapisovanja z velikostjo povečuje, s tem pa se povečuje tudi odzivni čas aplikacije[17].

2.5.2 Keychain

UserDefaults je odličen vmesnik za shranjevanje majhne količine podatkov, kot je uporabniško ime. Problem takega shranjevanja je lahka dostopnost do podatkov. Ti podatki niso varno shranjeni, zato so lahka tarča zlonamerne napada.

Apple je za ta namen naredil programski vmesnik keychain services, ki skrbi za kriptirano shranjevanje občutljivih podatkov, kot so geslo, številka bančne kartice itd. Preko programskega vmesnika Keychain services lahko dodajamo, pridobivamo, brišemo ali spreminjamo shranjene podatke[7, 13].

Poglavje 3

Razvoj mobilne aplikacije

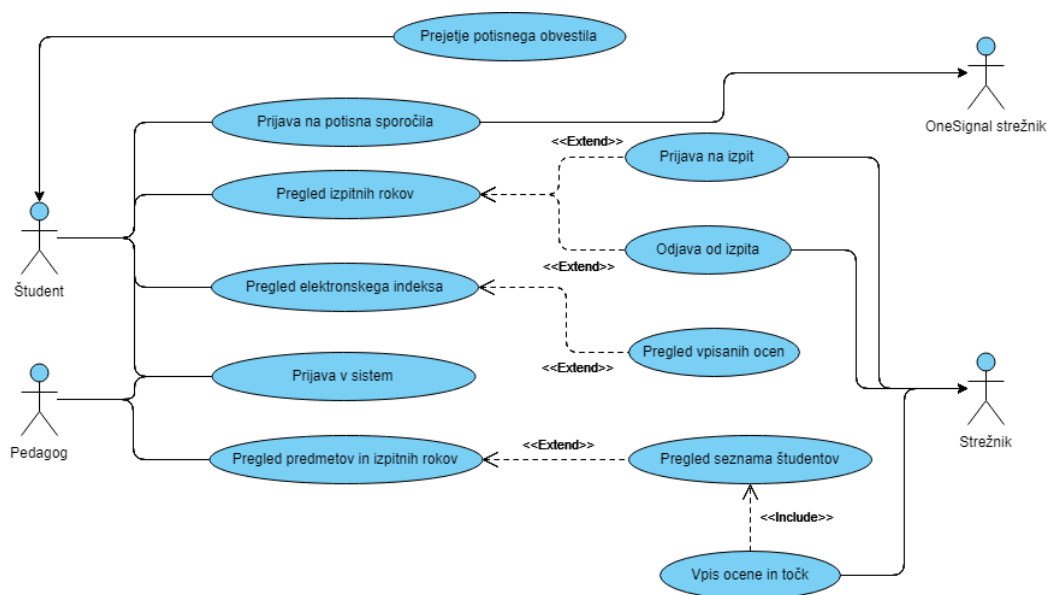
3.1 Prototip mobilne aplikacije mStudis

Za diplomsko nalogo sem si zadala narediti mobilno aplikacijo, ki bi bila namenjena tako študentom kot pedagogom (profesorjem in asistentom) Fakultete za računalništvo in informatiko. Mobilna aplikacija bi študentom omogočala prijavo k izpitom ter seznanitev z doseženim rezultatom, profesorji in asistenti pa bi lahko preverjali, kateri študentje so prijavljeni k posameznemu izpitu in koliko jih je.

Mobilno aplikacijo, ki sem jo poimenovala mStudis, sem implementirala za operacijski sistem iOS. Mobilna aplikacija je še prototip (beta različica), zato so baza, uporabniki in njihovi podatki izmišljeni in ne predstavljajo realnega stanja na Fakulteti za računalništvo in informatiko ter so le v pomoč pri realizaciji mobilne aplikacije in njeni predstavitvi. Aplikacija je na voljo je le na platformi TestFlight.

3.1.1 Diagram primerov uporabe

Diagram primerov uporabe (angl. Use-Case Diagram) je grafični prikaz interakcij med uporabnikom mobilne aplikacije in različnimi primeri uporabe, ki jih lahko uporabnik izvaja. Diagram primerov uporabe je zelo pomembno orodje za razvoj sistemov, saj omogoča pregled sistema na lahko razumljiv



Slika 1: Diagram primerov uporabe

način in je primeren tako za razvijalce sistema kot za uporabnike. Narejen je v jeziku UML (ang. Unified Modeling Language), ki je standardni zapis za modeliranje objektov in sistemov v resničnem svetu.

Diagram prikazuje primere uporabe mobilne aplikacije mStudis za akterja študenta ter pedagoga. Za uspešni zagon mobilne aplikacije mora imeti uporabnik nameščeno mobilno aplikacijo na svoji iOS napravi. Po zagonu se uporabniku odpre prijavno okno, kjer vnese svoje uporabniško ime in geslo.

V primerih, da geslo ni pravilno, uporabnik ne obstaja ali pa mobilna aplikacija nima dostopa do strežnika, se uporabniku izpiše opozorilo, da je prišlo do napake.

V primeru pravilne prijave se uporabniško ime in geslo pošljeta na strežnik, ki vrne podatke uporabnika. Če je uporabnik študent, se vrnejo podatki o imenu in priimku in vpisni številki, seznam predmetov, ki jih obiskuje, seznam aktualnih izpitov ter seznam vseh ocen, ki jih je dosegel v času študija. Če pa je uporabnik pedagog, dobi poleg svojega imena in priimka še seznam aktualnih izpitov ter za vsakega izmed izpitov še seznam prijavljenih

šudentov.

Po uspešni prijavi se mobilna aplikacija z uporabniškim ID-jem prijavi na potisna obvestila preko strežnika OneSignal .

Pregled izpitnih rokov študentu prikaže seznam vseh aktualnih rokov. Preko tega seznama se lahko študent prijavi ali odjavi od izpita. Vsak izpit ima unikatnen ID, ki se ob prijavi ali odjavi skupaj s podatki o študentu pošlje na strežnik, kjer se shrani v podatkovno zbirko. Študentu se ob tem spremeni videz prijavnega oziroma odjavnega gumba, tako da odraža stanje v bazi.

Študent ima na voljo tudi pregled elektronskega indeksa, kjer lahko vidi seznam vseh vpisanih ocen v času študija.

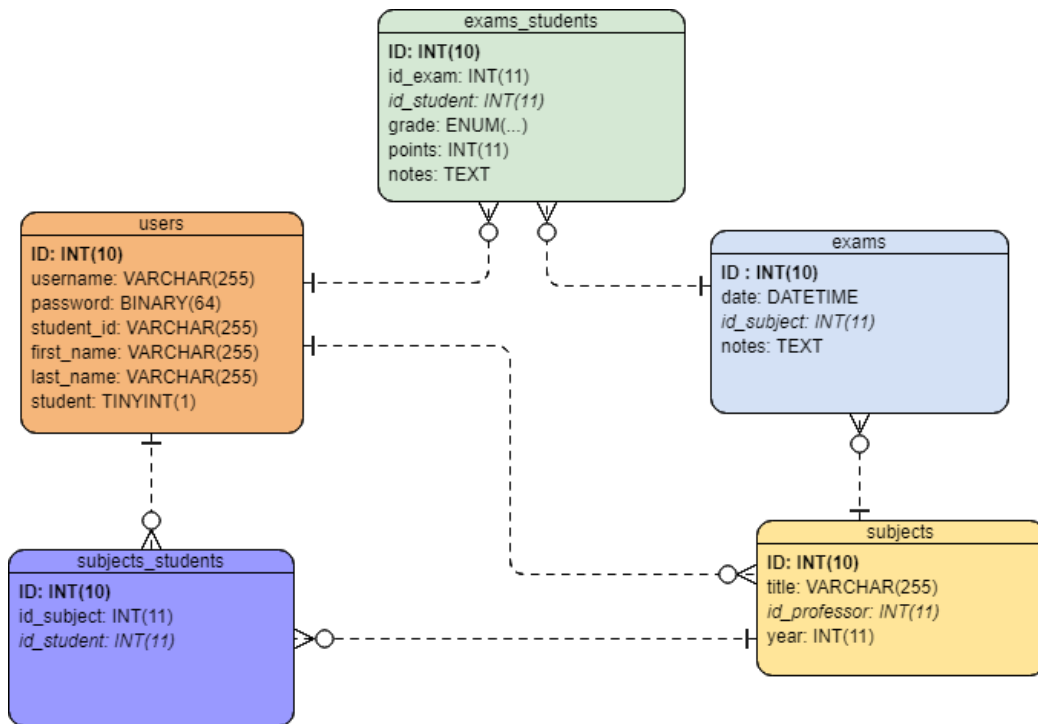
Pedagogu se po prijavi prikaže seznam vseh aktualnih izpitov. Poleg osnovnih podatkov o izpitu se mu zraven prikaže tudi število prijavljenih študentov. Z izbiro predmeta se mu odpre poimenski seznam študentov, prijavljenih na izpit. V primeru, da na izpit ni prijavljen noben študent, se mu izpiše ustrezno opozorilo.

V pregledu seznama študentov, prijavljenih na izpit, lahko pedagog, profesor oziroma asistent, vpiše tudi doseženo oceno in točke. Ob pritisku na gumb Shrani se vpisani podatki pošljejo na strežnik. Poleg uporabniškega ID-ja študenta in ID-ja izpita se po kriptirani povezavi zraven v datoteki JSON pošljeta še ocena in število točk. V primeru, da imamo v podatkovni zbirki že vpisano oceno za dotičnega študenta pri tem predmetu, se mu ocena prepíše. Po shranjeni oceni pošlje strežnik ID uporabnika in ime predmeta na OneSignal strežnik, ki pošlje dotičnemu študentu potisno obvestilo o vpisani oceni.

3.1.2 Podatkovni model

Predstavljen podatkovni model vsebuje 5 tabel: `users`, `exams_students`, `exams`, `subjects` in `subjects_students`.

Tabela **users** predstavlja vse uporabnike - študente, profesorje in asistente. V njej so za vsakega uporabnika shranjeni podatki:



Slika 2: Podatkovni model

- **ID** primarni ključ, obvezen
- **Username** uporabniško ime uporabnika, obvezen
- **Password** uporabniško geslo, ki je v bazo shranjeno s hashem gesla SHA-512, obvezen
- **Student_id** vpisna številka študenta, neobvezen
- **First_name** ime uporabnika, obvezen
- **Last_name** priimek uporabnika, obvezen
- **Student** vrednost, ki pove, ali je uporabnik študent, obvezen

Tabela **subjects** predstavlja vse predmete, ki se izvajajo na fakulteti. V testno podatkovno bazo sem za potrebe testiranja in predstavitev mobilne

aplikacije mStudis dodala nekaj naključnih predmetov in jih porazdelila v različne letnike. V tej tabeli se za vsak predmet hranijo podatki:

- **ID** primarni ključ, obvezen
- **Title** naziv predmeta, obvezen
- **Id_professor** ID profesorja, ki izvaja predmet, tuji ključ, obvezen
- **Year** letnik, v katerem se izvaja predmet, obvezen

Tabela **subjects_students** predstavlja študente in predmete, ki jih obiskujejo. V tej tabeli se hranijo podatki:

- **ID** primarni ključ, obvezen
- **Id_subject** ID predmeta, ki ga obiskuje študent, tuji ključ, obvezen
- **Id_student** ID študenta, ki obiskuje predmet, tuji ključ, obvezen

Tabela **exams** predstavlja vse izpitne roke, ki so razpisani. V njej se za vsak izpit hranijo podatki:

- **ID** primarni ključ, obvezen
- **Date** datum in ura izpita, obvezen
- **Id_subject** ID izpitnega predmeta, tuji ključ, obvezen
- **Notes** dodatne informacije, ki jih profesor lahko doda k informaciji o izpitu (npr. učilnica, v kateri poteka izpit, ali so dovoljeni zapiski itd), neobvezen

Tabela **exams_students** predstavlja vse študente, ki so prijavljeni na izpitni rok. V primeru, da se študent preko mobilne aplikacije mStudis prijavi na izpit, se v tabelo doda nov vnos, če se študent odjavi, pa se mu pod oceno vpiše VP (vrnjena prijavnica). Ocena je podatkovnega tipa ENUM za katerega imamo vnaprej definirano zalogo vrednosti (ocene od 1-10, VP). Ob

vpisu ocen se vsakemu prijavljenemu študentu zraven izpitnega roka vpiše ocena in število točk. S tem hranimo zgodovino ocen in izpitnih rokov za vsakega študenta. V podatkovni tabeli `exams_students` se za vsak par izpit-študent hranijo podatki:

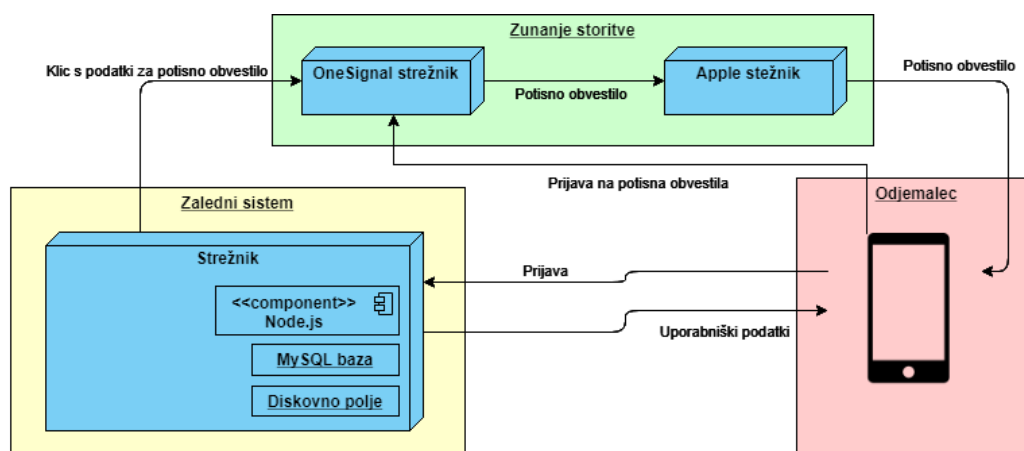
- **ID** primarni ključ, obvezen
- **Id_exam** ID izpita, ki ga je študent opravljal, tuji ključ, obvezen
- **Id_student** ID študenta, ki je izpit opravljal, tuji ključ, obvezen
- **Grade** dosežena ocena pri izpitu, neobvezen
- **Points** dosežene točke izpita, neobvezen
- **Notes** dodatne informacije o doseženi oceni, neobvezen

Testna podatkovna baza narejena za prototip mobilne aplikacije `mStudis` se nahaja na spletnem strežniku. Narejena je bila z orodjem za grajenje poizvedb (angl.: query builder) `Knex` in napolnjena s testnimi podatki, ki zajemajo 42 uporabnikov in 39 predmetov.

3.2 Arhitektura prototipa mobilne aplikacije `mStudis`

Mobilna aplikacija `mStudis` uporablja arhitekturo odjemalec-strežnik z zunanjimi spletnimi storitvami. Na sliki imamo štiri ključne komponente, od tega imata odjemalec in zaledni del vsak po eno, dve pa sta pri zunanjih storitvah. Povezave med komponentami so prikazane s puščicami, ki predstavljajo kriptirane povezave.

Odjemalec je iOS naprava, na kateri se nahaja prototip mobilne aplikacije `mStudis`. Aplikacija preko kriptirane povezave komunicira z zalednim sistemom in mu pošilja ter od njega prejema podatke. Zunanje storitve uporablja za prijavo in prejemanje potisnih obvestil.



Slika 3: Arhitektura mobilne aplikacije mStudis

V prototipni fazi je zaledni del narejen kot podpora za testiranje in predstavitev odjemalca oziroma mobilne aplikacije. Sestavljen je iz spletnega strežnika spisanega v Node.js okolju z ogrodjem za usmerjanje zahtevkov Express. Baza je MySQL, poizvedbe pa se gradijo s Knex.js. Komunikacija s strežnikom poteka preko protokola HTTPS. Podatki, ki se pošiljajo na strežnik in s strežnika, so v formatu JSON.

Prototip mobilne aplikacije mStudis uporablja dva zunanja sistema, ki ju aplikacija potrebuje za potisna obvestila. OneSignal strežnik skrbi za hrambo podatkov o uporabnikih, ki so naročeni na potisna obvestila, in pošiljanje potisnih obvestil Applovemu strežniku, ki jih nato dejansko pošlje na pravo napravo oziroma aplikacijo uporabnika. Več o potisnih obvestilih in delovanju knjižnice OneSignal je napisano v poglavju 3.4.

3.3 Implementacija prototipa mobilne aplikacije mStudis

Mobilna aplikacija deluje na operacijskem sistemu iOS 11 ali več. Za delovanje potrebuje internet preko brezžične povezave Wi-Fi ali preko mobilnega

omrežja. Za samo delovanje ne potrebuje dovoljenj, vendar uporabnikom priporočam, da za boljšo uporabniško izkušnjo dovolijo dostop do potisnih sporočil (push notifications).

Aplikacijo je mogoče naložiti na iPhone in iPad. Zasnovana je tako, da se prilagaja višini in širini zaslona, ter s tem ohranja svoj videz. To sem dosegla z uporabo Xcode funkcije Auto Layout, ki dinamično prilagodi položaje in velikost vseh pogledov glede na velikost zaslona.

Mobilna aplikacija je zasnovana tako, da se čim bolj približa grafični podobi Fakultete za računalništvo in informatiko oziroma Univerze v Ljubljani[5, 6]. Tako dominira rdeče-bela kombinacija barv, v ozadju pa so svetlo sivi krogi, ki jih najdemo tudi pri spletni strani Studis[16]. Mobilno aplikacijo je mogoče z majhnimi spremembami prilagoditi tudi drugim fakultetam.

Za logotip aplikacije sem prevzela logotip uporabljen pri spletni aplikaciji Studis[16]. Tako je logotip bel krog na rdečem ozadju, na sredini kroga pa je logotip Fakultete za računalništvo in informatiko. Pri prenosu logotipa iz spletne aplikacije v mobilno, sem ugotovila, da je ta slabo viden in ne predstavi pomena aplikacije, zato predlagam, da se ga v prihodnje spremeni.



Slika 4: Zaslona telefona iPhone z ikono aplikacije mStudis

Aplikacija mStudis se odpre s pritiskom na ustrezno ikono.

3.3.1 Prijava v sistem

Po zagonu mobilne aplikacije se uporabniku prikaže obrazec za vpis uporabniškega imena in gesla. V prototipni različici ima mobilna aplikacija mStudis enega uporabnika pedagoga in enega uporabnika študenta. Predlagam, da se v produkcijskem okolju uporabijo uporabniška imena in gesla študentov, profesorjev in asistentov iz Active Directory, ki jih že sedaj uporabljajo za prijavo v spletno aplikacijo Studis[16].

V primeru, da uporabniško ime ali geslo nista pravilna, da uporabnik nima interneta ali pa je prišlo do kakšne druge napake, se uporabniku izpiše opozorilo o napaki.

Za lažji dostop ob naslednjih prijavah se po prvi prijavi uporabniško ime in geslo shranita v keychain. Tako uporabniku ob zagonu ni treba vsakič vpisovati uporabniškega imena in gesla. Za še lažjo prijavo v mobilno aplikacijo sem naredila tudi avtomatično prijavo, ki se izvede, ko uporabnik zažene aplikacijo brez da bi se predhodno iz nje odjavil.

Po prijavi v mobilno aplikacijo se uporabniku odpre forma, katerega osrednji del je namenjen prikazovanju podatkov, v zgornjem delu je na levi strani logotip Fakultete za računalništvo in informatiko, na desni strani pa navzdol obrnjena puščica, preko katere lahko uporabnik dostopa do kontekstnega menija. V kontekstnem meniju se najprej izpišejo osnovni podatki uporabnika (ime in priimek ter vpisna številka študenta oziroma ime in priimek profesorja oziroma asistenta), nato pa še opcije menija, ki se sproti prilagajajo glede na to, ali je uporabnik v mobilno aplikacijo prijavljen kot študent ali kot pedagog ter v katerem delu mobilne aplikacije se trenutno nahaja.

Nadaljnja uporaba mobilne aplikacije je odvisna od tega, ali se uporabnik v mobilno aplikacijo prijavlja kot študent ali kot pedagog.



Slika 5: Obrazec za vpis uporabniškega imena in gesla

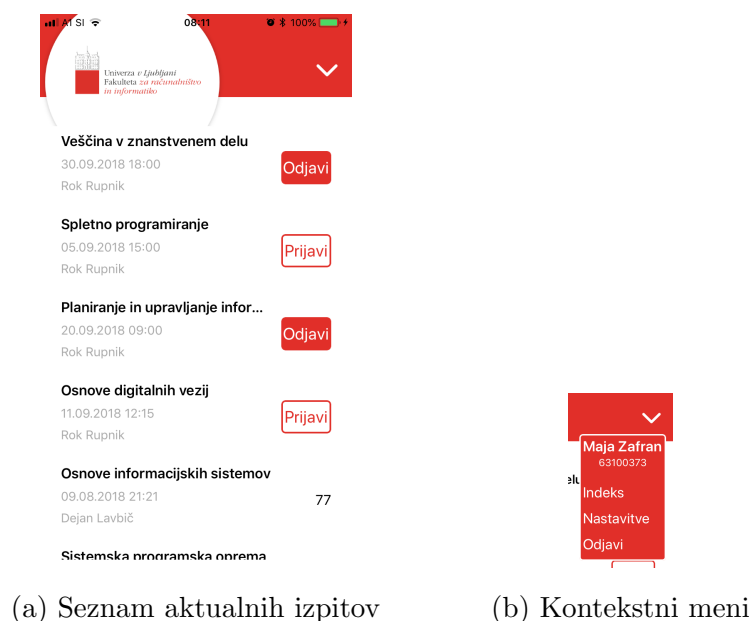
3.3.2 Modul namenjen študentom

Pregled izpitnih rokov

Glavna funkcionalnost mobilne aplikacije, ki je namenjena študentom, je upravljanje prijave na izpite, zato se študentu po prijavi v mStudis takoj odpre seznam njegovih aktualnih izpitov. Za vsak izpit se izpišejo naziv predmeta, datum in ura opravljanja izpita ter nosilec predmeta. Za izpite, h katerim študent še lahko pristopi, se na desni strani nahaja ukazni gumb za prijavo k izpitu. Če pa je na izpit že prijavljen, se na istem mestu pojavi ukazni gumb, preko katerega se lahko odjavi od izpita. V primeru, da je izpit že opravljal in ga je nosilec tudi že ocenil, se na tem mestu prikaže število doseženih točk.

V tem delu mobilne aplikacije lahko študent v kontekstnem meniju izbira med opcijami Indeks, Nastavitve in Odjavi. Želeni prikaz izbere s pritiskom na ustrezno besedo.

Po izbiri opcije Indeks se študentu odpre okno elektronskega indeksa, po izbiri opcije Nastavitve se odpre okno, kjer lahko upravlja s svojimi nastavi-



Slika 6: Pregled izpitnih rokov za študenta

tvami, z izbiro opcije Odjavi pa se odjavi iz mobilne aplikacije mStudis.

Pregled elektronskega indeksa

Za študenta je pregled ocen ena od pomembnejših informacij. Zato sem v mobilno aplikacijo mStudis dodala tudi ta pregled.

Po izbiri opcije kontekstnega menija Indeks se na zaslonu prikaže seznam opravljenih izpitov, ločenih po posameznem letniku. Za vsak predmet so vidni podatki o nazivu, oceni in nosilcu predmeta ter o datumu opravljanja izpita.

Na spletni strani Studis[16] se poleg teh osnovnih podatkov o opravljenem predmetu izpišejo še nekateri drugi podatki, kot so številka predmeta, kreditne točke itd. Ti podatki za namen te aplikacije niso bistveni, zato jih vanjo nisem vključila. Vsekakor se v prihodnosti lahko dodajo, če se za to izkaže potreba ali želja.

V tem delu mobilne aplikacije lahko študent v kontekstnem meniju izbira med opcijami Izpiti, Nastavitve in Odjava. Želeni prikaz izbere s pritiskom



Slika 7: Pregled ocen za študenta

na ustrezno besedo.

Po izbiri opcije Izpiti se študentu odpre okno s pregledom vseh aktualnih izpitov, po izbiri opcije Nastavitve se odpre okno, kjer lahko upravlja s svojimi nastavitvami, z izbiro opcije Odjava pa se odjavi iz mobilne aplikacije mStudis.

3.3.3 Modul namenjen pedagogom

Pregled predmetov in izpitnih rokov

Po prijavi v mobilno aplikacijo se profesorju oziroma asistentu prikaže seznam njegovih tekočih izpitov. Za vsak izpit se izpiše naziv predmeta ter datum in ura opravljanja izpita, na desni strani pa se mu izpiše število na izpit prijavljenih študentov. S pritiskom na posamezni izpit se pedagogu odpre seznam vseh prijavljenih študentov. V primeru, da na izpit ni prijavljen noben študent, se mu izpiše ustrezno obvestilo, seznam prijavljenih študentov pa se ne odpre.



(a) Seznam tekočih izpitov

(b) Kontekstni meni

Slika 8: Pregled predmetov in izpitov za profesorja oziroma asistenta

V tem delu mobilne aplikacije lahko pedagog v kontekstnem meniju izbira med opcijami Nastavitve in Odjavi. Želeni prikaz izbere s pritiskom na ustrezno besedo.

Po izbiri opcije Nastavitve se odpre okno, kjer lahko upravlja s svojimi nastavitvami, z izbiro opcije Odjavi pa se odjavi iz mobilne aplikacije mStudis.

Pregled seznama študentov

Poleg tega, da lahko profesor oziroma asistent preko mobilne aplikacije mStudis pregleduje, kateri študentje so prijavljeni na izpit iz njegovega posameznega predmeta, ima tudi možnost, da za vsakega študenta posebej vpiše še doseženo število točk in temu ustrezno oceno ali pa ga odjavi iz izpita.

Ko pedagog na seznamu njegovih tekočih izpitov izbere posamezni izpit, se mu odpre seznam imen in priimkov ter vpisnih števil študentov. Na desni strani sta pri vsakem študentu dve vnosni polji. Levo polje je namenjeno vpisu dosežene ocene, desno pa vpisu na izpitu doseženih točk. Če profesor

Veščina v znanstvenem delu

Jana Jesenek 63121234	ocena	točke
Tine Tim 63121234	ocena	točke
Klementina Kranjc 63121234	ocena	točke
Tilen Tratnik 63121234	ocena	točke
Sandi Smolej 63121234	ocena	točke
Živa Groza 63121234	ocena	točke
Matej Munc 63121234	ocena	točke
Jerneja Jereb 63121234	ocena	točke

Za vrneno prijavo vpiši "vp".

Shrani

Slika 9: Seznam prijavljenih študentov

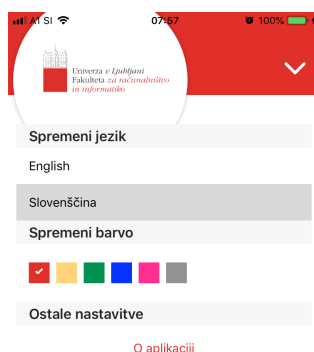
ali asistent namesto številke v katerokoli polje vpiše črki VP (vrnjena prijavnica), bo s tem študentu s seznama »vrnil«
prijavnico za izpit oziroma ga od izpita odjavil.

Pedagog ima kadarkoli možnost, da vpisane ocene in točke (ali odjavo od izpita) shrani s pritiskom na ukazni gumb Shrani, ki se nahaja v spodnjem desnem kotu. Po shranitvi se vpisani podatki shranijo v bazo, študentu pa se avtomatično pošlje potisno obvestilo o vpisanih podatkih. Za vrnitev nazaj na seznam izpitov se v desnem zgornjem kotu namesto kontekstnega menija nahaja ukazni gumb Nazaj. Vsi podatki, ki niso bili shranjeni, se po kliku na ukazni gumb Nazaj pobrišejo.

3.3.4 Nastavitve

Mobilna aplikacija je razvita tako, da se lahko prilagodi želeni celostni podobi. Tako jo je možno hitro spremeniti, če bi jo želela uporabljati še katera druga fakulteta .

Uporabnik lahko v mobilni aplikaciji trenutno spreminja jezik in barvo.

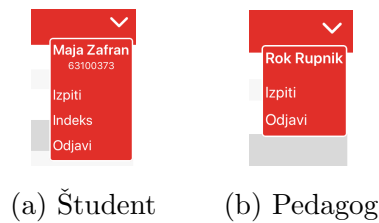


Slika 10: Nastavitve

Glede jezika ima uporabnik na voljo slovenščino, ki je prednastavljeno že izbrana, in angleščino. Z izbiro angleškega jezika lahko zato mobilno aplikacijo mStudis uporabljajo tudi gostujoči študentje, profesorji in asistenti. Možno je nastaviti tudi prevladujočo barvo mobilne aplikacije, ki je na začetku nastavljena na rdečo, ker je ta barva prevladujoča barva celostne podobe Fakultete za računalništvo in informatiko. Rdeča barva logotipa ostane nespremenjena ne glede na spremembo prevladujoče barve mobilne aplikacije.

Vsaka sprememba jezika ali barve se avtomatično shrani za nadaljnjo uporabo.

V tem delu mobilne aplikacije lahko uporabnik v kontekstnem meniju izbira med opcijami Izpiti, Indeks in Odjava za uporabnika študenta oziroma med opcijami Izpiti in Odjava za uporabnika pedagoga. Želeni prikaz izbere s pritiskom na ustrezno besedo.



Slika 11: Kontekstna menija glede na vrsto uporabnika

3.4 Potisna obvestila

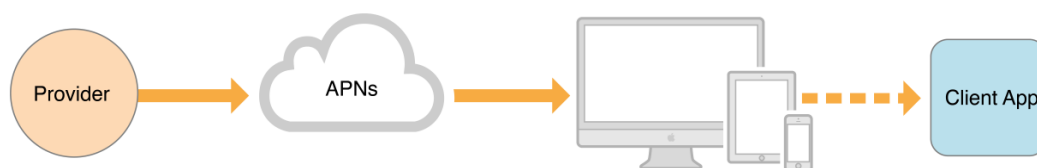
Velika prednost aplikacij na mobilnih napravah je možnost implementacije potisnih obvestil. Potisna obvestila omogočajo uporabniku prejetje obvestil iz priljubljenih aplikacij ne glede na to, ali je ta odprta, v ozadju ali pa celo neaktivna. Večina najbolj znanih aplikacij, kot so Facebook, Instagram, Twitter, e-pošta itd., preko potisnih obvestil obvešča uporabnika o dogajanju na njegovem računu. Tako uporabniku ni potrebno odpirati aplikacije samo za to, da vidi njemu pomembne novosti.

Apple Push Notification Service ali APNs je Applova platforma, ki omogoča razvijalcem aplikacij za iOS operacijski sistem, da v svoje aplikacije implementirajo prejetje potisnih obvestil. Platforma je izšla leta 2009 skupaj z iOS sistemom 3.0 in je bila kasneje dodana tudi za operacijski sistem Mac OS X.

Potisna obvestila se lahko uporabljajo v različne namene: za obveščanje uporabnika o novicah, za obvestila o prejetem sporočilu ali klicu v aplikaciji. Lahko vsebujejo zvok, slike in/ali prilagojeno vsebino.

Implementacija potisnih obvestil je sestavljena iz dveh delov. Prvi del implementacije poteka preko Apple portala za razvijalce, kjer je treba aplikacijo registrirati za potisna sporočila in pridobiti ustrezen kriptiran certifikat.

Drugi del pa se implementira v kodi aplikacije. Ob zagonu aplikacije se na Applovo platformo za potisna obvestila pošlje unikaten in za aplikacijo specifičen žeton ter se vzpostavi kriptirana in obstojna IP povezava. Žetoni so referenca Applove storitve potisnih obvestil, kateri napravi in kateri aplikaciji na napravi naj sistem dostavi potisno obvestilo. Razvijalci aplikacije morajo



Slika 12: Dostava potisnega obvestila od mobilne aplikacije[1]

zato skupaj z obvestilom na Applovo platformo poslati le pravilen žeton, APNs pa nato poskrbi za pravilno razpošiljanje takšnega obvestila.

Potisna obvestila so lahko v obliki sporočila, zvoka in/ali posodobitve značke na ikoni aplikacije. Oblika potisnega obvestila je skupaj z ostalimi informacijami, ki jih želimo poslati uporabniku, zapakirana v JSON slovar in poslana v telesu HTTP/2 sporočila. Največja dovoljena velikost sporočila je odvisna od tipa potisnih sporočil: če je potisno sporočilo tipa VOIP, je maksimalna velikost 5KB, za navadna potisna sporočila pa je 4KB. Potisna obvestila, ki presegajo maksimalno velikost, so avtomatično zavrnjena.

Prejem potisnih obvestil na telefon oziroma aplikacijo ne pomeni nujno tudi prikaz teh obvestil. To je prepuščeno razvijalcem aplikacije, ki se odločijo, kako naj se prejeto potisno obvestilo prikaže in kako naj se aplikacija obnaša ob prejemu le-tega. Najbolj pogosti odziv, ki sem ga implementirala tudi v svoji aplikaciji mStudis, je prikaz lokalnega obvestila.

V mobilni aplikaciji mStudis sem uporabila knjižnico OneSignal za lažje upravljanje s potisnimi obvestili. Mobilno aplikacijo sem najprej registrirala na njihovi spletni strani, kamor sem namestila tudi primerne Appleove certifikate. V mobilno aplikacijo sem dodala prikazno okno za dovoljenje uporabnika za prikaz potisnih obvestil. V primeru, da jih uporabnik dovoli uporabljati, pošlje mobilna aplikacija uporabniški ID na spletni vmesnik OneSignal. Vsak uporabnik je tako enolično shranjen s svojim uporabniškim ID-jem, kar nam omogoča pošiljanje potisnega sporočila točno določenemu uporabniku. Implementacijo potisnih obvestil sem dodala tudi na strežnik, tako da ob vpisani oceni študent avtomatično prejme potisno obvestilo. To se mu prikaže na vrhu zaslona, ne glede na to, ali je mobilna aplikacija odprta,

v ozadju ali neaktivna. Z izbiro potisnega sporočila se mobilna aplikacija mStudis odpre, prijavi študenta in prikaže seznam tekočih izpitnih rokov ter vpisanih ocen.

Potisna obvestila bi lahko uporabili še za marsikaj, kot na primer opomnik študenta, da se bliža rok prijave na izpit, opomnik, kdaj ali čez koliko časa se bo izpit začel, obvestilo, kje poteka izpit in podobno. Prav tako bi lahko naredili potisna obvestila za pedagoge, preko katerih bi bili obveščeni, na primer, koliko študentov je prijavljenih na izpit ali bi dobili opomnik, da se bo izpit kmalu začel.

V prototipu mobilne aplikacije sem želela nakazati možnost, ki jo ponuja Apple operacijski sistem, in sem tako implementirala le pošiljanje potisnega obvestila v primeru vnosa ocene[9, 12, 10].

Poglavje 4

Zaključek

Cilj moje diplomske naloge je bil razviti prototip mobilne aplikacije mStu-dis, ki omogoča študentom ter profesorjem in asistentom lažji pregled nad aktualnimi izpiti.

Pri diplomski nalogi sem se soočila z nekaterimi izzivi, ki mi jih je uspelo rešiti. Največji izziv mi je predstavljala postavitve baze in testnih podatkov, ki sem jih potrebovala za prikaz mobilne aplikacije. S podatkovnimi bazami in postavljanjem strežnika pred pisanjem diplomske naloge nisem imela izkušenj. Strežnik sem najprej postavila lokalno, ob izdaji mobilne aplikacije na TestFlight, pa sem strežnik in bazo postavila na splet za lažji dostop. Prav tako mi je nekaj preglavic povzročala implementacija potisnih obvestil, zato sem se za implementacijo le teh odločila uporabiti knjižnico OneSignal. Ta je zelo poenostavila prijavo na potisna obvestila in pošiljanje le teh, saj njihov strežnik skrbi za komunikacijo z Applovim sistemom za pošiljanje potisnih obvestil.

Z diplomsko nalogo sem pridobila veliko dodatnih izkušenj ne samo glede snovanja mobilnih aplikacij temveč tudi njihovega razvoja. V veliko pomoč mi je bila uradna Applova dokumentacija za programski jezik Swift, ki je zelo dobro napisana.

4.1 Možnosti za nadaljevanje razvoja

V diplomski nalogi sem se osredotočila na razvoj mobilne aplikacije na iOS operacijskem sistemu. Seveda poznamo tudi druge operacijske sisteme kot so na primer Android, Windows Mobile, Blackberry OS, itd., katerih uporabniki se zagotovo najdejo med študenti ter profesorji in asistenti na fakulteti. Tako bi bil moj prvi predlog za izboljšavo izdelava aplikacij za druge mobilne operacijske sisteme in s tem povečanje števila uporabnikov aplikacije.

Tudi sama aplikacija za operacijski sistem iOS ima veliko možnosti za izboljšavo. Ena izmed manj zahtevnih dodatnih funkcionalnosti bi bil opomnik na prihajajoči izpit. Tega bi se lahko implementiralo lokalno preko lokalnih obvestil ali pa preko potisnih obvestil s strani strežnika. Prav tako bi lahko uporabnik dobil potisna obvestila ob spremembi lokacije, ure ali dneva izpita ter z datumom in uro vpogleda izpita.

Večja in bolj kompleksna možnost izboljšave je možnost vpisa študenta v višji letnik. Za realizacijo le-te bi se naredilo novo okno s formo za vpis v višji letnik. Študentu bi bila ta forma na voljo samo v primeru, da bi izpolnjeval vse vpisne pogoje. Forma bi bila predizpolnjena, kjer bi bilo to mogoče. Študent bi tako podatke pregledal in po potrebi dopolnil, nato pa bi se ta forma poslala naprej v obdelavo.

Prav tako koristna izboljšava za študente bi bila komunikacija s študentskim referatom in zaposlenimi na fakulteti. Ta bi lahko v prvi fazi potekala preko e-pošte, kjer bi se uporabniku z izbiro ustreznega ukaza odprla njegova aplikacija za e-pošto, ali pa bi se pošiljanje le-te implementiralo kar preko vmesnika mobilne aplikacije. Kasneje bi se lahko dodalo še komuniciranje preko pogovora v vmesniku mobilne aplikacije.

Literatura

- [1] APNs Overview. Dosegljivo: <https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html>. [Dostopano 11. 8. 2018].
- [2] App Review. Dosegljivo: <https://developer.apple.com/app-store/review/>. [Dostopano 22. 7. 2018].
- [3] App Store. Dosegljivo: <https://www.apple.com/si/ios/app-store/>. [Dostopano 22. 7. 2018].
- [4] Beta Testing Made Simple. Dosegljivo: <https://developer.apple.com/testflight/>. [Dostopano 22. 7. 2018].
- [5] CELOSTNA GRAFIČNA PODOBA. Dosegljivo: <https://fri.uni-lj.si/sl/celostna-graficna-podoba>. [Dostopano 30. 6. 2018].
- [6] CELOSTNA GRAFIČNA PODOBA. Dosegljivo: https://www.uni-lj.si/medijsko_sredisce/celostna_graficna_podoba/. [Dostopano 30. 6. 2018].
- [7] Keychain services.
- [8] News and Updates. Dosegljivo: <https://developer.apple.com/news/releases/>. [Dostopano 22. 7. 2018].
- [9] Notifications. Dosegljivo: <https://developer.apple.com/notifications/>. [Dostopano 11. 8. 2018].

-
- [10] OneSignal. Dosegljivo: <https://onesignal.com/>. [Dostopano 11. 8. 2018].
 - [11] OneSignal iOS SDK. Dosegljivo: <https://github.com/OneSignal/OneSignal-iOS-SDK>. [Dostopano 22. 7. 2018].
 - [12] Product Overview. Dosegljivo: <https://documentation.onesignal.com/docs>. [Dostopano 11. 8. 2018].
 - [13] Storing Keys in the Keychain. Dosegljivo: https://developer.apple.com/documentation/security/certificate_key_and_trust_services/keys/storing_keys_in_the_keychain. [Dostopano 22. 7. 2018].
 - [14] Swift 4. Dosegljivo: <https://developer.apple.com/swift/>. [Dostopano 21. 7. 2018].
 - [15] The Basics. Dosegljivo: <https://docs.swift.org/swift-book/LanguageGuide/TheBasics.html>. [Dostopano 21. 7. 2018].
 - [16] Študijski informacijski sistem. Dosegljivo: <https://studij.fri.uni-lj.si>. [Dostopano 1. 9. 2018].
 - [17] UserDefaults. Dosegljivo: <https://developer.apple.com/documentation/foundation/userdefaults>. [Dostopano 22. 7. 2018].
 - [18] What Is Core Data? Dosegljivo: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CoreData/index.html>. [Dostopano 22. 7. 2018].
 - [19] Xcode. Dosegljivo: <https://itunes.apple.com/us/app/xcode/id497799835>. [Dostopano 21. 7. 2018].
 - [20] Xcode 10. Dosegljivo: <https://developer.apple.com/xcode/>. [Dostopano 21. 7. 2018].

-
- [21] Xcode 10. Dosegljivo: <https://developer.apple.com/xcode/>. [Dostopano 21. 7. 2018].
- [22] Xcode IDE. Dosegljivo: <https://developer.apple.com/xcode/ide/>. [Dostopano 21. 7. 2018].
- [23] About Swift. Dosegljivo: <https://swift.org/about/>, 2015. [Dostopano 21. 7. 2018].
- [24] Number of mobile connections worldwide from 2008 to 2020 (in billions). Dosegljivo: <https://www.statista.com/statistics/371828/worldwide-mobile-connections/>, 2015. [Dostopano 22. 7. 2018].
- [25] Number of apps available in leading app stores as of 1st quarter 2018. Dosegljivo: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>, 2018. [Dostopano 22. 7. 2018].
- [26] Paul Solt. Swift vs. Objective-C: 10 reasons the future favors Swift. Dosegljivo: <https://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html>, 2015. [Dostopano 21. 7. 2018].